**الگوی Bridge یا پل**

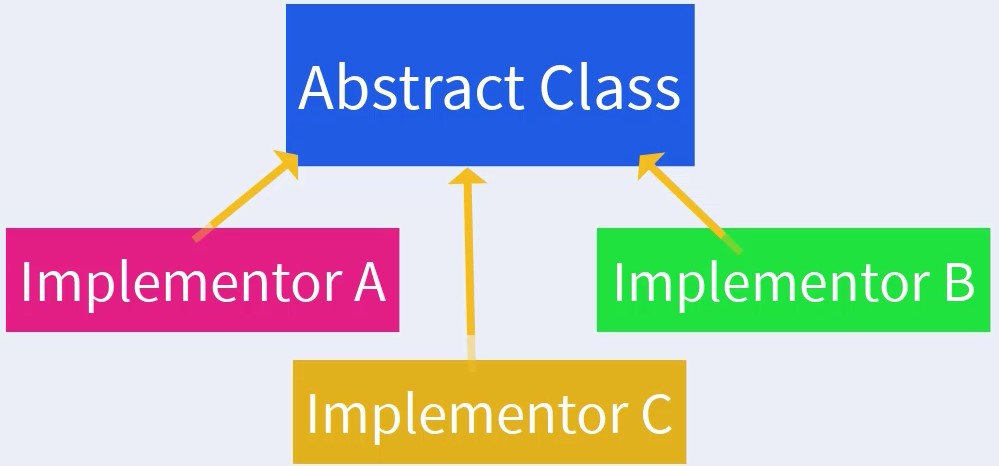
**هدف پیاده سازی:**

**جداسازی انتزاع و پیاده سازی**

**یادآوری: اگر بخواهیم برای یک کلاس چندین پیاده سازی را ارائه دهیم:**

**ایجاد یک کلاس Abstract و به ازای این کلاس چندین پباده سازی میتوانیم ایجاد کنیم که این کلاس ها همگی از abstract ارث بری میکنند.**

**مشکل این روش این است که همه کلاس های پیاده سازی شده اتصال محکمی به کلاس abstract ما دارند ( یعنی هر متدی به کلاس abstract اضافه شوند ،کلاس های پیاده سازی باید آن متد را پیاده سازی کنند.هرکدام آن را پیاده سازی نکنند برنامه compile نمی شود و نمی توان abstract را مستقل از پیاده سازی ها توسعه داد.)**

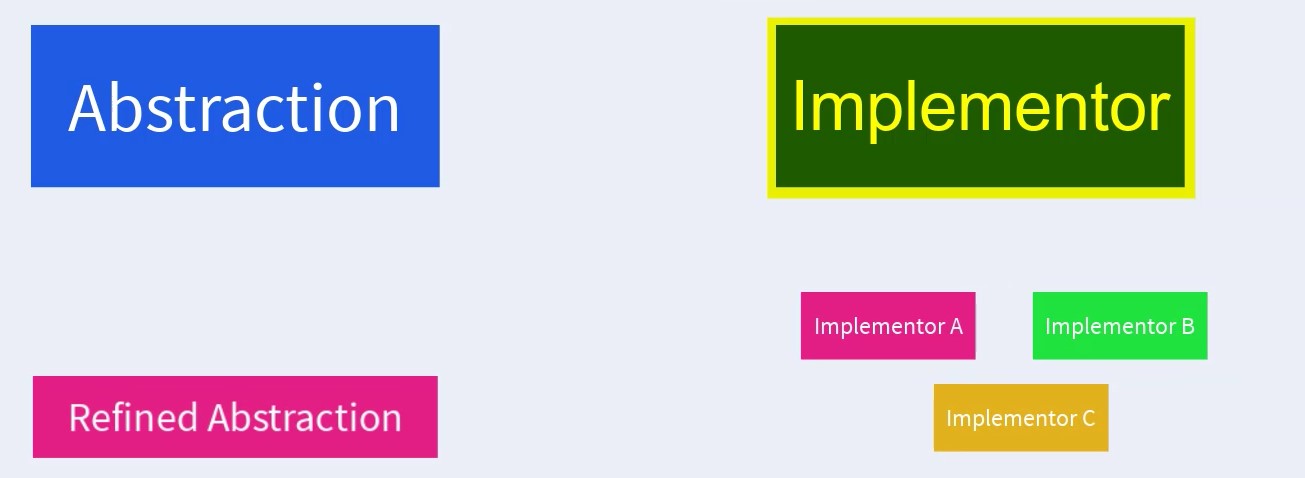
****

**حال الگوی Bridge کلاس abstract را به دو دسته ی abstraction و implementor تقسیم میکنند.(هر دو کلاس abstract هستند)**

کلاس **abstraction برای نمای بیرونی کلاس همان متد ها و property ها ( client می بیند.)**

**کلاس Implementor یا bridge برای پیاده سازی های داخلی کلاس(پیاده سازیabstraction )**

**و از کلاس Implementor یا bridge می توانیم چندین پیاده سازی داشته باشیم.**



Refiend Abstraction کلاسی ارث بری شده از abstraction است که به دلیل اینکه نمیتوانیم از کلاس abstraction نمونه ایجاد کرد از این کلاس استفاده می شود.